quattor

Quattor Quick Installation Guide

Version 1.0.x

EDMS id:	**Id**
Date:	**November 30, 2004**
Partner(s):	**Universidad Autónoma de Madrid**
Lead Partner:	**CERN**
Document status:	**FINAL**
Author(s):	Rafael A. García Leiva (angel.leiva@uam.es) German Cancio Meliá (German.Cancio@cern.ch) Piotr Poznański (Piotr.Poznanski@cern.ch)
File:	**quick-install-guide**

Abstract: *This document contains step by step instructions to install a basic quattor server for testing purposes.*

CONTENTS

1. INTRODUCTION

Quattor is a system administration tool suite providing a powerful, portable and modular tool for the automated installation, configuration and management of machines running UNIX derivatives like Linux and Solaris. For more information about quattor, and to freely download the latest software release, check the quattor web page at `http://quattor.org`.

1.1. OBJECTIVES OF THIS DOCUMENT

This guide provides step by step instructions to install a basic quattor server. This guide is intended for those who want to test the quattor software, or use quattor to manage a small site. Please refer to the *Quattor Installation and User Guide* document (check the links under the ’*documentation*’ tab at quattor web page) for instructions to how to install a production quality quattor server, and how to use the quattor software.

1.2. APPLICATION AREA

This document applies to the quattor tool suite (see `http://quattor.org`) versions 1.0.x. Please, mind that this document is **not** a manual for European DataGrid, LCG or EGEE testbeds installation.

1.3. ACRONYMS AND ABBREVIATIONS

The quattor-specific acronyms and abbreviations used in this guide are the following:

AII: Automated Installation Infrastructure

CAF: Common Application Framework

CCM: Configuration Cache Manager

CDB: Configuration Database

HLD: High Level Description

HLDL: High Level Description Language

NCM: Node Configuration Manager

NVA-API: Node View Access API

LLD: Low Level Description

LLDL: Low Level Description Language

NBP: Network Bootstrap Program

SPM: Software Package Management

SPMA: Software Package Management Agent

SWRep: Software Repository

OS	Version	Platform	Status
RedHat	7.3	i386	Supported
Fedora	Core 2	i386	Supported
RedHat Adv. Server	3	i386	Supported
RedHat Adv. Server	3	IA64	Supported
Scientific Linux	3.03	i386	Supported
Scientific Linux	3.03	IA64	Supported
Solaris	9	Sun	In Progress

Table 1: Supported Platforms

2. OVERVIEW OF QUATTOR INSTALLATION

A minimal installation of quattor needs at least a server node running RedHat Linux. All the quattor software packages can be freely downloaded from the quattor web page. However, quattor uses many other services for its operation. Among these services, we can mention the basic servers: Web, DNS, CVS and SSH, but depending on your quattor installation you may need FTP or NFS as well. And for the automatic installation of client nodes you need DHCP, TFTP and PXELinux. All those dependencies are mentioned in this installation guide when they are needed.

The primary platform for quattor is RedHat Linux, but platform independence is a design principle. Quattor has been tested in the platforms described on Table 1.

The examples and explanations provided in this guide assume a RedHat Linux 7.3 installation. Similar procedures should be used for other RedHat Linux versions and Solaris. Please refer to the *Quattor Installation and User Guide* for installation notes specific to each operating system and platform.

For software packages names, we use `<ver>` and `<arch>` tags instead of version numbers and architectures. For example, instead of

```
pan-1.0.6-2.i386.rpm
```

we will use

```
pan-<ver>.<arch>.rpm
```

For each package, install the latest version included in the quattor repository (see quattor web page) for the a given platform and quattor release.

As a general rule, you can assume that when a quattor software package installs a configuration file directly under `/etc`, this means that the default configuration values should work for most of the environments. However, if the package installs the configuration files under a documentation directory, this means that the default values do not work, and that you should edit and change the configuration parameters.

3. INSTALLING THE CONFIGURATION DATABASE

Install a RedHat distribution (including the Apache Web Server), and then, install the following quattor/CDB software packages:

```
rpm -ivh perl-LC-<ver>.<arch>.rpm
rpm -ivh perl-AppConfig-caf-<ver>.<arch>.rpm
rpm -ivh perl-CAF-<ver>.<arch>.rpm
rpm -ivh panc-<ver>.<arch>.rpm
rpm -ivh cdb-<ver>.<arch>.rpm
```

Enable the *Configuration Database Manager* account by changing its password:

```
passwd cdb
```

Create and initialize the configuration database by running, as user `root`, the CDB initialization script:

```
cdb-setup
```

Install the CDB server local management utility:

```
rpm -ivh cdb-simple-cli-<ver>.<arch>.rpm
```

The utility `cdb-simple-cli` should be used by the CDB manager (user `cdb`). For example, you can check that CDB is working properly with:

```
su - cdb cdb-simple-cli --list
```

The above command prints out all the templates stored on CDB. By the moment it should print something like:

```
[INFO]: listing templates ...
```

and nothing more since we do not have any configuration template loaded yet.

4. INSTALLING A SOFTWARE REPOSITORY

Install the following SWRep software packages:

```
rpm -ivh swrep-libs-<ver>.<arch>.rpm
rpm -ivh swrep-server-<ver>.<arch>.rpm
```

Copy SWRep client configuration file example:

```
cp /usr/share/doc/swrep-server-<ver>/swrep-server.conf /etc/swrep/
```

Edit the example file and modify the following configuration parameters:

- `name` and `owner` of the repository, used in configuration templates (see Section 5.),
- `url:` URL from where clients can download the software packages in the repository

Install the SWRep management client software:

```
rpm -ivh swrep-client-<ver>.<arch>.rpm
```

Copy SWRep client configuration file example:

```
cp /usr/share/doc/swrep-client-<ver>/swrep-client.conf /etc/swrep
```

Edit the example file and modify the following configuration parameters:

- `repository`: repository location in `user@host` format, where `user` is the `username` of the repository manager, for example we can use `swrep@host.example.org`,

The repository management is performed via the Secure Shell SSH. Add the SSH public key of the user that is going to manage the repository to the file:

```
/var/swrep/.ssh/authorized_keys
```

For example, if we want to use the user `user1` at host `host.example.org` to manage the repository, we should add something like the following to the `authorized_keys`:

```
environment="SSH_USER=user1" ssh-rsa AAAAB3NzaC1
YwP0jGlWvMl6D+LVIpCEU+qGhQdCNL691gvufcaK9jKwdfPB
VwlEbq8BpumdCOJlH3v8Q9i2hEqgCTT6WeRVxk+NCL5dQ6Ag
au1UpZ40= user1@host.example.org
```

Create your own key with the `ssh-keygen(1)` utility. It is very important that you specify in the file the `SSH_USER` environment variable as well.

Edit the `/etc/swrep/swrep.acl` file, and add a new entry with the user defined above:

```
user1:/
```

4.1. LOADING SOFTWARE PACKAGES

Start an SSH agent with:

```
ssh-agent $SHELL
ssh-add
```

Create a new platform to store RedHat 7.3 packages for the x86 architecture, with:

```
swrep-client addplatform i386_rh73
```

And inside this platform, we have to create the following areas:

```
swrep-client addarea i386_rh73 /base
swrep-client addarea i386_rh73 /updates
swrep-client addarea i386_rh73 /quattor
```

Copy the RedHat distribution software packages into the software repository, for example with (repeate for each distribution CDROM):

```
cp /mnt/cdrom/RedHat/RPMS/*.rpm /var/www/html/swrep/i386_rh73
```

And then bootstrap them into SWRep with:

```
swrep-client bootstrap i386_rh73 /base
```

Do exactly the same with the set of software packages updates, by downloading them from any RedHat mirror, and the quattor software packages (including the external packages list), by donwloading them from the quattor web page.

5. CREATING A CONFIGURATION PROFILE

The next step is to load into CDB those templates (written in Pan language) that compose the configuration profiles of our client nodes. Install the following RPM package:

```
rpm -ivh pan_templates-<ver>-noarch.rpm.
```

This RPM install the set of templates under the directory:

```
/usr/share/doc/pan_templates-<ver>
```

In order to load a template into CDB you can use the local management utility `cdb-simple-cli`. For example, to load a template called `my_template.tpl`, you can use (as `cdb` user):

```
cdb-simple-cli --add my_template.tpl
```

Load the following list of templates:

```
standard/pro_declaration_type_validation_functions
standard/pro_declaration_structure_validation_functions
standard/pro_declaration_types
standard/pro_declaration_structures
standard/pro_declaration_units
standard/pro_declaration_functions_filesystem
standard/pro_declaration_functions_network
standard/pro_declaration_functions_general
standard/pro_declaration_profile_base
```

In case of a test environment the templates for hardware description are not relevant. Load the following list of templates:

```
site_specific/pro_hardware_card_nic_intel_e100
site_specific/pro_hardware_ram_1024
site_specific/pro_hardware_harddisk_STD_80
site_specific/pro_hardware_cpu_GenuineIntel_Pentium_4_2600
site_specific/pro_hardware_asus_terminator_p4_533a
```

Create the template describing your software repository. Be carefull about the name of the template, that must be the name of the repository. For example with:

```
swrep-client template i386_rh73 > repository_foo_1_repository_i386_rh73
```

Then, load the resulting template an the following ones:

```
site_specific/pro_software_packages_redhat_7_3
site_specific/pro_software_packages_quattor
```

Finally, edit the following template, change your repository name, and load it:

```
site_specific/pro_software_foo_linux_1_1_0
```

Load the templates for the core configuration components:

```
components/pro_declaration_component_grub
components/pro_declaration_component_spma
components/pro_declaration_component_ntpd
components/pro_declaration_component_interactivelimits
components/pro_declaration_component_cron
components/pro_declaration_component_accounts
components/pro_declaration_component_aii
components/pro_software_component_grub
components/pro_software_component_spma
components/pro_software_component_ntpd
components/pro_software_component_interactivelimits
components/pro_software_component_cron
components/pro_software_component_accounts
components/pro_software_component_accounts_sysgroups
components/pro_software_component_accounts_sysusers
```

Edit the following AII template, and change at the OS Installation section, the fiels `server` and `cdb` with the hostname of the quattor server machine, and load the modified template:

```
components/pro_software_component_aii
```

Edit the following base template, and change at the network configuration information, and load the modified template:

```
site_specific/pro_system_base
```

Load also the template:

```
site_specific/pro_system_foo
```

Finally create a new template called `profile_hostname` where `hostname` is the name of the client node you want to install, use the `site_specific/profile_bar1` template as an example, and load the new template.

6. AUTOMATED INSTALLATION OF CLIENT NODES

Install the following software packages:

```
rpm -ivh perl-Compress-Zlib-<ver>.<arch>.rpm
rpm -ivh perl-XML-Simple-<ver>.<arch>.rpm
rpm -ivh ccm-<ver>.<arch>.rpm
rpm -ivh cdbsync-<ver>.<arch>.rpm
rpm -ivh aii-server-<ver>.<arch>.rpm
```

Edit the DHCP configuration example file:

```
/usr/share/doc/aii-server-<ver>/eg/dhcpd.conf
```

Insert the configuration parameters about your network (read the comments on the file), and copy the file to the `/etc` directory.

Create a new directory, called `/osinstall/nbp/rh73` and copy there the RedHat Linux bootstrap files, for example with:

```
cp /mnt/cdrom/images/pxeboot/* /osinstall/nbp/rh73
```

Create the following symbolic lynk:

```
ln -s /osinstall/ks /var/www/html/ks
```

And copy the contents of the RedHat installation CD-ROMs to a place readable from your web server, for example:

```
cp -r /mnt/cdrom/* /var/www/html/rh73
```

But, if you want to save space, you can reuse the packages of your repository, by creating a symbolic link:

```
ln -s /var/www/html/swrep/i386_rh73/ /var/www/html/rh73/RedHat/RPMS
```

Copy the acknowledgment cgi-bin script to your Apache web server cgi-bin directory:

```
cp /usr/sbin/aii-installack.cgi /var/www/cgi-bin
```

Add the following entry to the `/etc/sudoers` file:

```
apache your_hostname=(ALL) NOPASSWD: /usr/sbin/aii-shellfe
```

Edit the AII configuration file example file:

```
/usr/share/doc/aii-<ver>/eg
```

Modify the `cdburl` configuration parameter according to your needs, and move the file to its right place:

```
/etc/aii-shellfe.conf
```

In order to add the new node to be managed by AII, we have to use:

```
aii-shellfe --configure node_name
```

Once the new node has been configured, we have to mark the node for its complete installation with:

```
aii-shellfe --install node_name
```

Finally, switch on the client node, and it after a while it should be install and configure according to its profile.